Machine Learning Module

Decision Tree

Manu K. Gupta

MFS of data science and AI,

IIT Roorkee.

1st March, 2:00 PM - 4:00 PM

Session Plan

Decision Trees

- Entropy
- Gini Index

Decision Tree Regression

Decision Trees

Background

- What type of problems do we solve using decision trees?

Background

- What type of problems do we solve using decision trees?
    - Supervised Learning: Classification tasks (typically).
    - Can be used for Regression also.
- Features $\Leftrightarrow$ Attribute.

Background

- What type of problems do we solve using decision trees?
  - Supervised Learning: Classification tasks (typically).
  - Can be used for Regression also.
- Features $\Leftrightarrow$ Attribute.
- Learned function is represented by a *decision tree*.

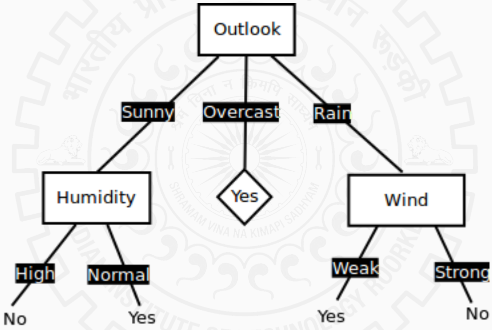
Background

- What type of problems do we solve using decision trees?
  - Supervised Learning: Classification tasks (typically).
  - Can be used for Regression also.
- Features $\Leftrightarrow$ Attribute.
- Learned function is represented by a *decision tree*.
- What is a decision tree?

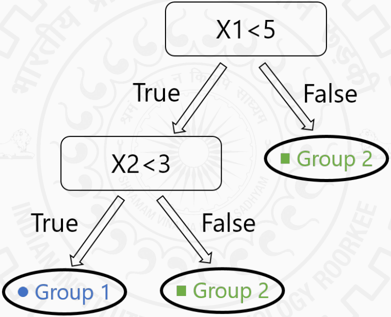
Background

- What type of problems do we solve using decision trees?
  - Supervised Learning: Classification tasks (typically).
  - Can be used for Regression also.
- Features $\Leftrightarrow$ Attribute.
- Learned function is represented by a *decision tree*.
- What is a decision tree?
  - A structure that includes a root node, branches, internal nodes and leaf nodes.
  - A set of decision rules.

Example 1


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Weak
Strong
No
Yes
Yes
No

Example 2

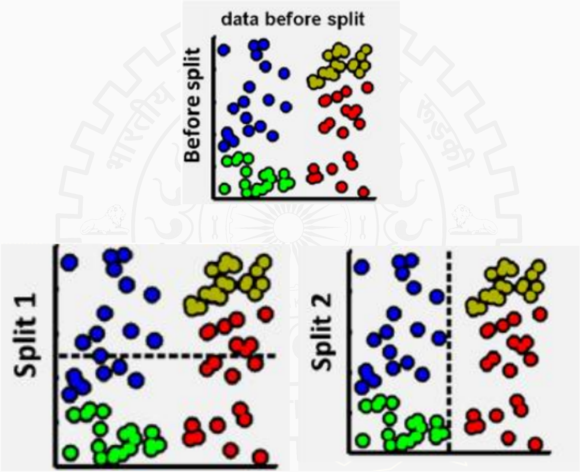


Major challenges in building decision tree?

- Ordering of features

Which order to choose?

Ordering of features



Major algorithms

- Iterative Dichotomiser 3 (ID3)
- Classification and Regression Tree (CART)

- Entropy function Vs Gini Index
- Information Gain Vs Gini Gain

Entropy

- Assume our data is set S with C many classes.
- p_c is the probability that a random element of S belongs to class c.
- Probability vector $p = [p_1, p_2, \cdots, p_C]$ is the class distribution of the set S
- Entropy of the set S

$$\boxed{H(S) = -\sum_{c \in C} p_c log_2 p_c}$$

- Example
  $S = \{1, 1, 0, 1, 0, 1\}, S_1 = \{0, 0, 0, 0, 0, 0\}, S_2 = \{0, 0, 0, 1, 1, 1\}$

Entropy

- Assume our data is set S with C many classes.
- p_c is the probability that a random element of S belongs to class c.
- Probability vector $p = [p_1, p_2, \cdots, p_C]$ is the class distribution of the set S
- Entropy of the set S

$$\boxed{H(S) = -\sum_{c \in C} p_c log_2 p_c}$$

- Example
  $S = \{1, 1, 0, 1, 0, 1\}, S_1 = \{0, 0, 0, 0, 0, 0\}, S_2 = \{0, 0, 0, 1, 1, 1\}$

- Some dominant classes $\Rightarrow$ Small entropy
- Equiprobable classes $\Rightarrow$ High entropy

- Some dominant classes $\Rightarrow$ Small entropy
- Equiprobable classes $\Rightarrow$ High entropy
- Can you use Entropy to order the features?

Information Gain

Entropy of S minus weighted sum of entropy of its children

$$\boxed{IG(S,F) = H(S) - \sum_{f \in F} \frac{|S_f|}{|S|} H(S_f)}$$

Choose an attribute with the largest information gain.

Decision Tree Algorithm

Step 1: Calculate Entropy of the target

Step 2: Find out information gain for each attribute.

Step 3: Choose attribute with the largest information gain as the decision node.

Step 4: A branch with entropy of 0 is a leaf node. A branch with entropy more than 0 needs further splitting.

Step 5: The ID3 algorithm runs recursively on the non-leaf branches, until all the data is classified.

A complete example

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Step 1: Calculate Entropy of the target

Step 1: Calculate Entropy of the target

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

$\boxed{\begin{aligned} \text{Entropy(PlayGolf)} &= \text{Entropy }(5,9) \\ &= \text{Entropy }(0.36, 0.64) \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94 \end{aligned}}$

Step 2: Find out information gain for each attribute

Step 2: Find out information gain for each attribute

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Step 2: Find out information gain for each attribute

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Information gain =0.94 - 0.693 = 0.247

Step 2 contd...

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Step 2 contd...

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

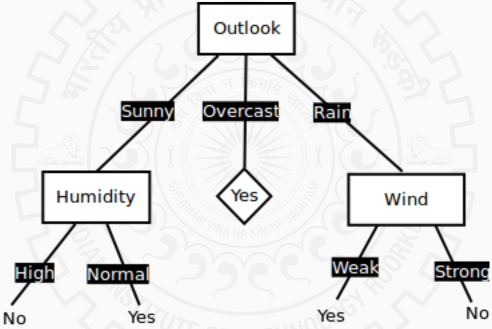
		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Step 3: Choose attribute with the largest information gain.

Step 4: A branch with entropy of 0 is a leaf node. A branch with entropy more than 0 needs further splitting.

Step 5: The ID3 algorithm runs recursively on the non-leaf branches, until all the data is classified.

Trained decision tree



Gini Index

- Purpose?

Gini Index

- Purpose?
- To identify the ordering of features in decision trees.

Gini Index

- Purpose?
- To identify the ordering of features in decision trees.
- Also known as Gini impurity
- CART (Classification and Regression Tree) algorithm

Gini Index

- Purpose?
- To identify the ordering of features in decision trees.
- Also known as Gini impurity
- CART (Classification and Regression Tree) algorithm
- Gini Index and Gini gain
- Similar mechanism as entropy and information gain in ID3 algorithm

- Calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly.

$$\boxed{\text{Gini Index } = 1 - \sum_{i=1}^{n} p_i^2}$$

where p_i denotes the probability of an element being classified for a distinct class.

- Examples: $S = \{0, 0, 1, 1\}$, $S_1 = \{0, 0, 0, 0\}$.

- Calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly.

$$\boxed{\text{Gini Index } = 1 - \sum_{i=1}^{n} p_i^2}$$

where p_i denotes the probability of an element being classified for a distinct class.

- Examples: $S = \{0, 0, 1, 1\}$, $S_1 = \{0, 0, 0, 0\}$.
- Gini index varies between 0 and 1.
- 0.5 Gini index implies equal distribution.
- 0 expresses the purity of classification.

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2: Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2:	Sunny	3	2	5
Outlook	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node)

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2:	Sunny	3	2	5
Outlook	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2:	Sunny	3	2	5
Outlook	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$
- Gini (Outlook = Sunny)

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2:	Sunny	3	2	5
Outlook	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$
- Gini (Outlook $=$ Sunny) $= [1 - (3/5)^2 - (2/5)^2] = 0.48$

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2: Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$
- Gini (Outlook = Sunny) $= [1 - (3/5)^2 - (2/5)^2] = 0.48$
- Gini(Outlook = Overcast) = 0 and Gini(Outlook = Rainy) = 0.48

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2: Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$
- Gini (Outlook = Sunny) $= [1 - (3/5)^2 - (2/5)^2] = 0.48$
- Gini(Outlook = Overcast) $= 0$ and Gini(Outlook = Rainy) $= 0.48$
- Gini(Children node)

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2: Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$
- Gini (Outlook = Sunny) $= [1 - (3/5)^2 - (2/5)^2] = 0.48$
- Gini(Outlook = Overcast) $= 0$ and Gini(Outlook = Rainy) $= 0.48$
- Gini(Children node) $= 5/14 * 0.48 + 4/14 * 0 + 5/14 * 0.48 = 0.3429$

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2: Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$
- Gini (Outlook = Sunny) $= [1 - (3/5)^2 - (2/5)^2] = 0.48$
- Gini(Outlook = Overcast) $= 0$ and Gini(Outlook = Rainy) $= 0.48$
- Gini(Children node) $= 5/14 * 0.48 + 4/14 * 0 + 5/14 * 0.48 = 0.3429$

Gini Gain

Gini Gain

$$\boxed{\text{Gini Gain} = \text{Gini(Parent node)} - \text{Gini(children node)}}$$

		Yes	No	Total
Feature 2: Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

- Gini (parent node) $= [1 - (10/14)^2 - (4/14)^2] = 0.4082$
- Gini (Outlook $=$ Sunny) $= [1 - (3/5)^2 - (2/5)^2] = 0.48$
- Gini(Outlook $=$ Overcast) $= 0$ and Gini(Outlook $=$ Rainy) $= 0.48$
- Gini(Children node) $= 5/14 * 0.48 + 4/14 * 0 + 5/14 * 0.48 = 0.3429$

Gini Gain $= 0.4082 - 0.3429 = 0.065$

Gini Index

- `sklearn.tree.DecisionTreeClassifier` will choose the attribute with the largest Gini Gain as the Root Node.
- A branch with Gini of 0 is a leaf node, while a branch with Gini more than 0 needs further splitting.

Gini Index

- `sklearn.tree.DecisionTreeClassifier` will choose the attribute with the largest Gini Gain as the Root Node.
- A branch with Gini of 0 is a leaf node, while a branch with Gini more than 0 needs further splitting.

Training Algorithm	CART	ID3
Metric	Gini Index	Entropy Function
Cost function	Minimize Gini Impurity	Largest Information gain

Demo

Demo

- DT basic demo
- DT demo tennis
- DT Bill Authentication
- Diabetes dataset
- Lead dataset

Practice the other demos with different data-sets on decision trees.

Decision Tree Regression

Decision Trees for Regression

Country	Rim	Tires	Type	Price
Japan	R14	195/60	Small	11.95
Japan	R15	205/60	Medium	24.76
Germany	R15	205/60	Medium	26.9
Germany	R14	175/60	Compact	18.9
Germany	R14	195/60	Compact	24.65
Germany	R15	225/60	Medium	33.2
USA	R14	185/75	Medium	13.15
USA	R14	205/75	Large	20.225
USA	R14	205/75	Large	16.145
USA	R15	205/70	Medium	23.04

Let's take an example of the price of car based on features like country, Rim, tires and type.

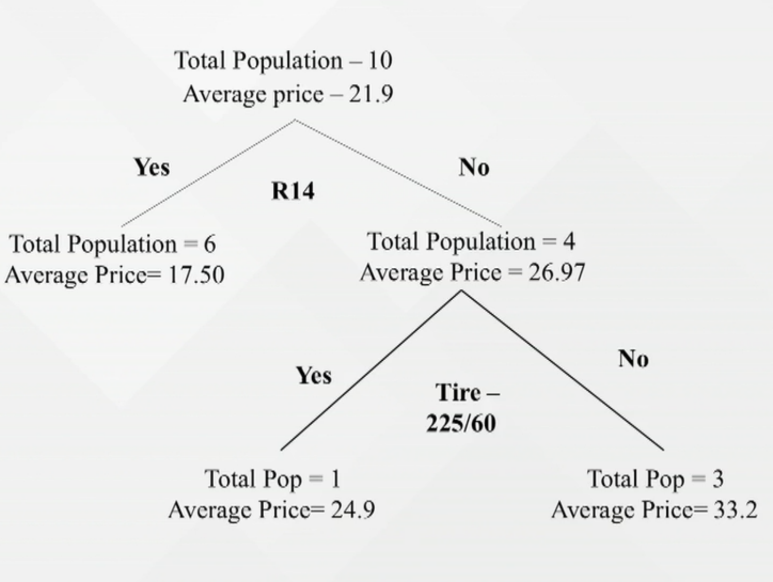
Decision Trees for Regression

Suppose we start splitting the tree using RIM Attribute

Decision Trees for Regression

Suppose we start splitting the tree using RIM Attribute

Decision Trees for Regression



Important points

- Feature splitting
- Prediction at a node is based on the average value of the target column
- MSE is used as a purity metric to decide the split
- MSE is the mean of the squared difference between the actual and the predicted value.
- Feature with the lowest MSE is chosen

MSE calculation

MSE calculations for RIM and country Germany

Calculate the MSE for each split of each feature

Rim - R14

Total Population =10
Average price = 21.9

Yes: Total Population = 6, Average Price= 17.50

Price	Pred.
11.95	17.50
18.9	17.50
24.65	17.50
13.15	17.50
20.22	17.50
16.14	17.50

$$MSE = \frac{1}{6}[(11 \cdot 9 - 17.5)^2 + (18 \cdot 9 - 17.5)^2 + \cdots (16 \cdot 4 - 17.5)^2] = 18.67$$

No: Total Population = 4, Average Price = 26.97

Price	Pred.
24.76	26.97
26.90	26..97
33.20	26.97
23.04	26.97

$$MSE = \frac{1}{4}[(24.76 - 26.97)^2 + (26.90 - 26.97)^2 + \cdots (23 \cdot 04 - 26.97)^2] = 14.78$$

Country - Germany

Total Population =10
Average price = 21.9

Yes: Total Population = 4, Average Price= 25.91

Price	Pred.
26.90	25.91
18.90	25.91
26.45	25.91
33.20	25.91

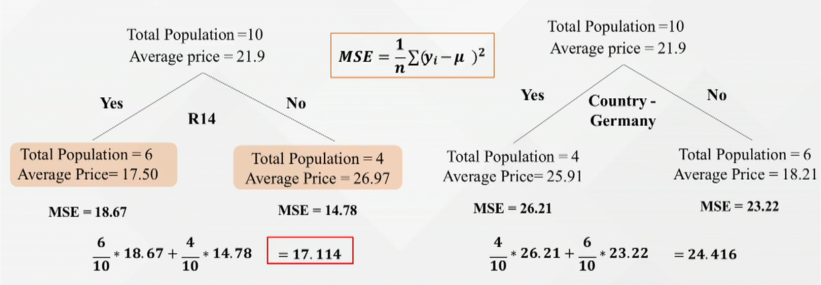
MSE =26.21

No: Total Population = 6, Average Price = 18.21

Price	Pred.
11.95	18.21
24.76	18.21
13.15	18.21
20.22	18.21
16.14	18.21
23.04	18.21

MSE =23.22

Calculate the weighted MSE for both the features



Feature with the lowest MSE is chosen for constructing the tree.

Hence Rim feature will be chosen to start splitting the tree.

Demo

Thank you!